



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER OF PATENTS AND TRADEMARKS  
Washington, D.C. 20231  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/409,613	10/01/1999	ARMIN HAROLD CHRISTOFFERSON	R09-99-091	5640

24033 7590 11/07/2002

KONRAD RAYNES VICTOR & MANN, LLP  
315 SOUTH BEVERLY DRIVE  
SUITE 210  
BEVERLY HILLS, CA 90212

EXAMINER

PHAM, HUNG Q

ART UNIT	PAPER NUMBER
----------	--------------

2172

DATE MAILED: 11/07/2002

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/409,613

Applicant(s)

CHRISTOFFERSON ET AL.

Examiner

HUNG Q PHAM

Art Unit

2172

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 09 September 2002.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-36 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-36 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on \_\_\_\_\_ is: a) ☐ approved b) ☐ disapproved by the Examiner.
- If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

**Priority under 35 U.S.C. §§ 119 and 120**

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
- a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892) 4) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_\_
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948) 5) ☐ Notice of Informal Patent Application (PTO-152)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) \_\_\_\_\_ 6) ☐ Other: \_\_\_\_\_

### **DETAILED ACTION**

1. Applicant's arguments with respect to claims 1-36 in the Request for Reconsideration received on 09/09/2002 have been considered but are moot in view of the new ground(s) of rejection.

#### ***Claim Rejections - 35 USC § 103***

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

**Claims 1-36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Harper [USP 5,765,165].**

Regarding to claims 1, 10, and 19, Harper teaches a method, system, and an article of manufacture for utilizing a hash function to speed up the average search for duplicates of file in a UNIX file system (Col. 1, lines 5-49). The Harper method could be used either to detect whether duplicates exist on a linked list or to detect whether an element to be added to a linked list is a duplicate of one which already exists on the

Art Unit: 2172

linked list (Col. 4, lines 21-29). As shown in FIG. 1-3C, the relationship between the data blocks, the inode, and a directory that references the file is illustrated (Col. 1, line 13-16). As shown in FIG. 8, in order to check whether an element such as a file to be added to a linked list is a duplicate of one already on the list, a hash function to hash the identifier of the added element to one bit position of the hash bit map at step 820 as the step of *applying a function to map the input file name to a value*. Harper does not explicitly disclose the step of *processing a data structure to determine whether there is a preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps, wherein two files that map to a same value according to the function are capable of having a same name*. However, Harper teaches that: a hash bit map as *a data structure* is initialized to 0 at step 800 and for each element on a linked list; the identifier for that element is hashed to one bit position of the hash bit map at step 810. The identifier of an element to be added to the linked list is also hashed to one bit location on the bit map at step 820, and that location is checked to determine if the value in the bit location is 1 or 0. If it is 0, no possible duplicate exists and the process returns. If the value of the bit position is 1, a possible duplicate exists (Col. 8, lines 20-35). Thus, through the process, at steps 800-810, the bit positions correspond to the elements as files in the linked list having identifiers as file names will have the bit value equal to 1. If the added element is hashed to a value corresponds to a location of the hash bit map, and the bit value of that location equal to 1 as being checked at step 830, a duplication of identifiers is possible. This technique illustrates the step of *processing a data structure to determine whether there is a preexisting file in the file system having a name*

*that maps, according to the function, to the same value to which the input file name maps, wherein two files that map to a same value according to the function are capable of having a same name.* Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method by processing a hash bit map as a data structure to determine a possible duplication of file name, and by using the step of processing, a duplication of file name in the file system could be identified quickly by scanning the hash bit map.

Regarding to claims 2, 11 and 20, Harper teaches all the claimed subject matters as discussed in claims 1, 10 and 19, Harper further discloses *the mapped-to values require fewer bits of storage than the file names* (Col. 4, lines 29-34).

Regarding to claims 3, 12 and 21, Harper teaches all the claimed subject matters as discussed in claims 1, 10 and 19, Harper further discloses *the function is a hash function that maps the input file name to an integer value, and wherein the data structure includes an entry for each possible integer value capable of being generated from the hash function* (Col. 2, line 57-Col. 3, line 27 and Col. 4, lines 29-34).

Regarding to claims 4, 13 and 22, Harper teaches all the claimed subject matters as discussed in claims 3, 10, 21, Harper further discloses the step of *determining whether the entry for the integer value to which the input file name maps indicates the*

*presence of one preexisting file mapping to the same integer value as the input file name* (Col. 8, lines 25-35).

Regarding to claims 5, 14 and 23, Harper teaches all the claimed subject matters as discussed in claims 4, 13 and 22, Harper further discloses *the data structure is a one-dimensional array and wherein each entry is capable of having one of two values, further comprising setting the entry to a first value if there is one preexisting file name in the file system that maps to the integer value for the entry, and wherein determining whether there is one preexisting file comprises determining whether the entry for the integer value to which the input file name maps has the first value* (Col. 5, lines 29-35 and Col. 8, lines 20-35).

Regarding to claims 6, 15 and 24, Harper teaches all the claimed subject matters as discussed in claims 1, 10, 19, Harper further discloses the steps of *applying the function to each file name in the file system to map each file name to one value; and indicating in the data structure, for each file name, that there is one preexisting file for the value to which the file name maps* (Col. 8, lines 20-28).

Regarding to claims 7, 16 and 25, Harper teaches all the claimed subject matters as discussed in claims 6, 15 and 24, Harper further discloses the step of *scanning each file in the file system to determine if there is at least one preexisting file having the same name as the input file name if there is one preexisting file in the file system having a name that*

*maps, according to the function, to the same value to which the input file name maps* (Col. 8, lines 29-35).

Regarding to claims 8, 17 and 26, Harper teaches all the claimed subject matters as discussed in claims 7, 16 and 25, Harper does not explicitly disclose the step of *adding the input file as a new file to the file system if no preexisting file in the file system has the same name as the input file name; and rejecting the access request if there is a preexisting file in the file system having the same name*. However, as shown in FIG. 8, when an element such as a file is added to the file system, the element identifier is checked for a possible duplicate and if there is no duplicate, the process return at step 850, otherwise the fact is reported to the calling process. Thus, the calling process is the process for adding a file to the file system and obviously, the Harper process of adding a file will *add the input file as a new file to the file system if no preexisting file in the file system has the same name as the input file name; and rejecting the access request if there is a preexisting file in the file system having the same name* by displaying a message as in most of conventional file system such as Window 95. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method by including the step of adding the input file and rejecting the input file in order to add a file to the file system.

Regarding to claims 9, 18 and 27, Harper teaches all the claimed subject matters as discussed in claims 7, 16 and 25, but fails to disclose the step of *updating a*

*preexisting file in the file system having the same name as the input file with the data in the input file if there is such a preexisting file; and rejecting the access request if there is no preexisting file in the file system having the same name as the input file name.* However, as shown in FIG. 8, when an element such as a file is added to the file system, the element identifier is checked for a possible duplicate and if there is no duplicate, the process return at step 850, otherwise the fact is reported to the calling process. In order to update a preexisting file, the Harper process could be modified by return to the calling process if a duplicate occurs and report to the calling process if there is no match. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method by including the step of updating if there is a preexisting file and rejecting if there is no preexisting file in order to update a file in a file system.

Regarding to claims 28, 31 and 34, Harper teaches all the claimed subject matters as discussed in claims 1, 10 and 19, Harper further discloses the step of *searching the file system for one preexisting file having the same name as the input file name if the data structure indicates that one preexisting file has a name that maps, according to the function, to the same value to which the input file maps; and performing an operation if the file system includes one preexisting file having the same name as the input file* (Col. 8, lines 25-35).



Regarding to claims 29, 32 and 35, Harper teaches all the claimed subject matters as discussed in claims 28, 31 and 34, Harper does not explicitly disclose the step of *applying update data to the preexisting file having the same name as the input file if the file system includes one preexisting file having the same name as the input file*. However, as shown in FIG. 1, Harper teaches that the file name "proposal" is associated with an inode number pointing to the inode containing information about the file "proposal." The inode number "248231" points to inode 24823 which contains information about the file "proposal" such as when the file was created, and when it was last modified. The inode points to data blocks. The data blocks contain the actual characters in the file such as the textual contents of the "proposal" after which the file was named (Col. 1, lines 40-49). Thus, if the calling process is an updating process, and there is a preexisting file having the same name as the input file after comparing the identifier with identifier of each element on the list (Col. 8, lines 25-35), obviously, the updated data such as date, and the actual data of the file will be applied to the preexisting file. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method by including the step of applying update data to the preexisting file in order to update a file in a file system.

Regarding to claims 30, 33 and 36, Harper teaches all the claimed subject matters as discussed in claims 28, 31, 34, Harper does not explicitly disclose the steps of *returning an error if the file system includes one preexisting file having the same name as the input file; and adding the input file to the file system if the file system does not include one*

*preexisting file having the same name as the input file.* However, as shown in FIG. 8, when an element such as a file is added to the file system, the element identifier is checked for a possible duplicate and if there is no duplicate, the process return at step 850, otherwise the fact is reported to the calling process. Thus, the calling process is the process for adding a file to the file system. And obviously, the step of adding the file to the file system will be executed if there is no duplicate, otherwise, the fact is reported as the step of returning an error. Therefore, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the Harper method by using the step of returning an error or adding the input file name in order to add a file to a file system.

### **Conclusion**

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hung Pham whose telephone number is 703-605 4242. The examiner can normally be reached on Monday-Friday, 7:00 Am - 3:30 Pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, VU, KIM YEN can be reached on 703-305 4393. The fax phone numbers for the organization where this application or proceeding is assigned are 703-746 7239 for regular communications and 703-746 7238 for After Final communications. Any

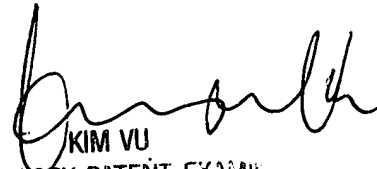
Application/Control Number: 09/409,613

Page 10

Art Unit: 2172

inquiry of a general nature or relating to the status of this application or proceeding  
should be directed to the receptionist whose telephone number is 703-305 3900.

Examiner: Hung Pham  
October 28, 2002

  
KIM VU  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100